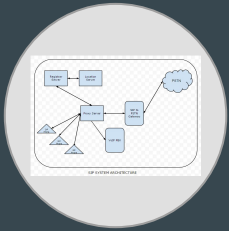


Swiftly Salesforce

...

Salesforce REST API, iOS Swift, Mobile SDK

Highlight Points



Architecture



Mobile App



REST API



Authentication



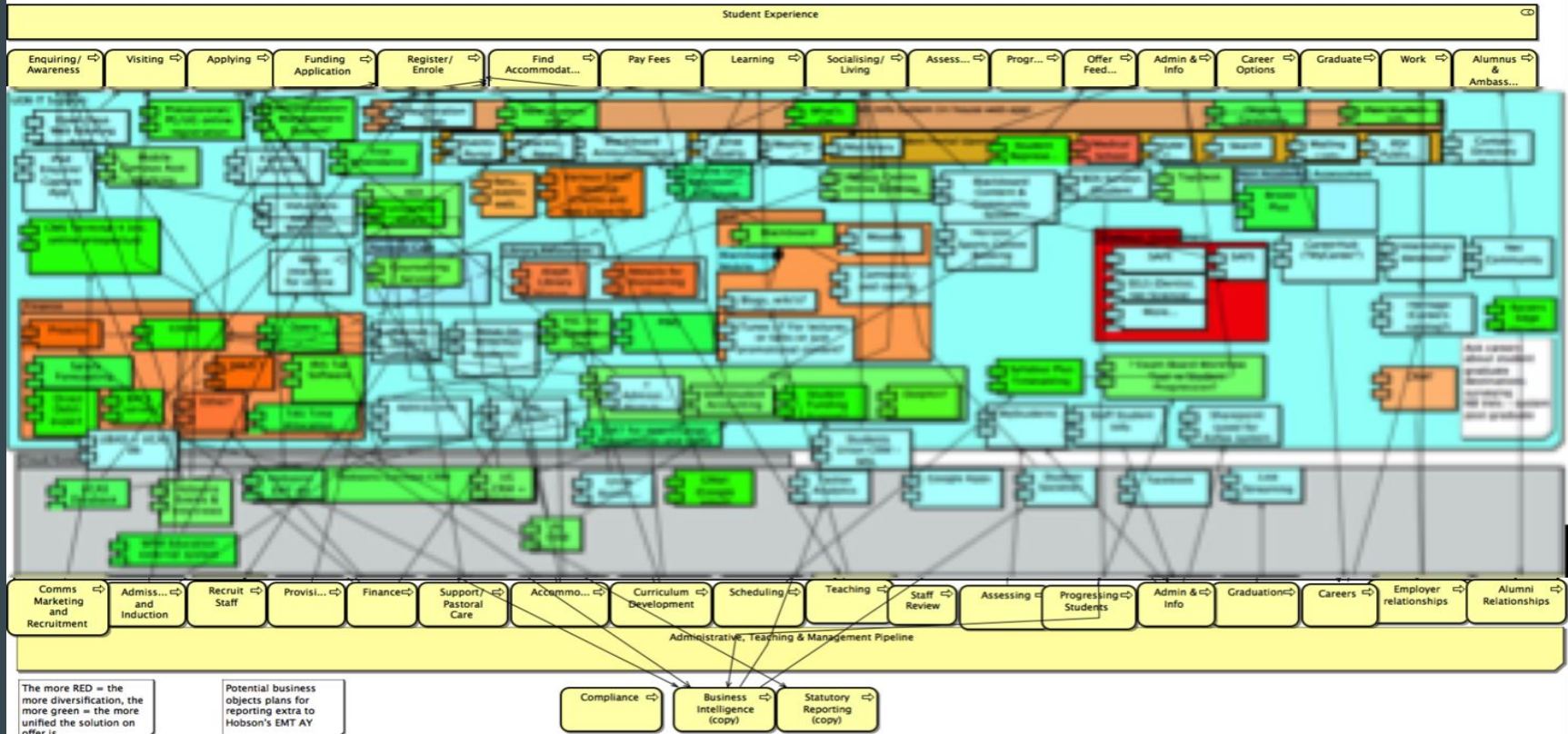
Mobile SDK



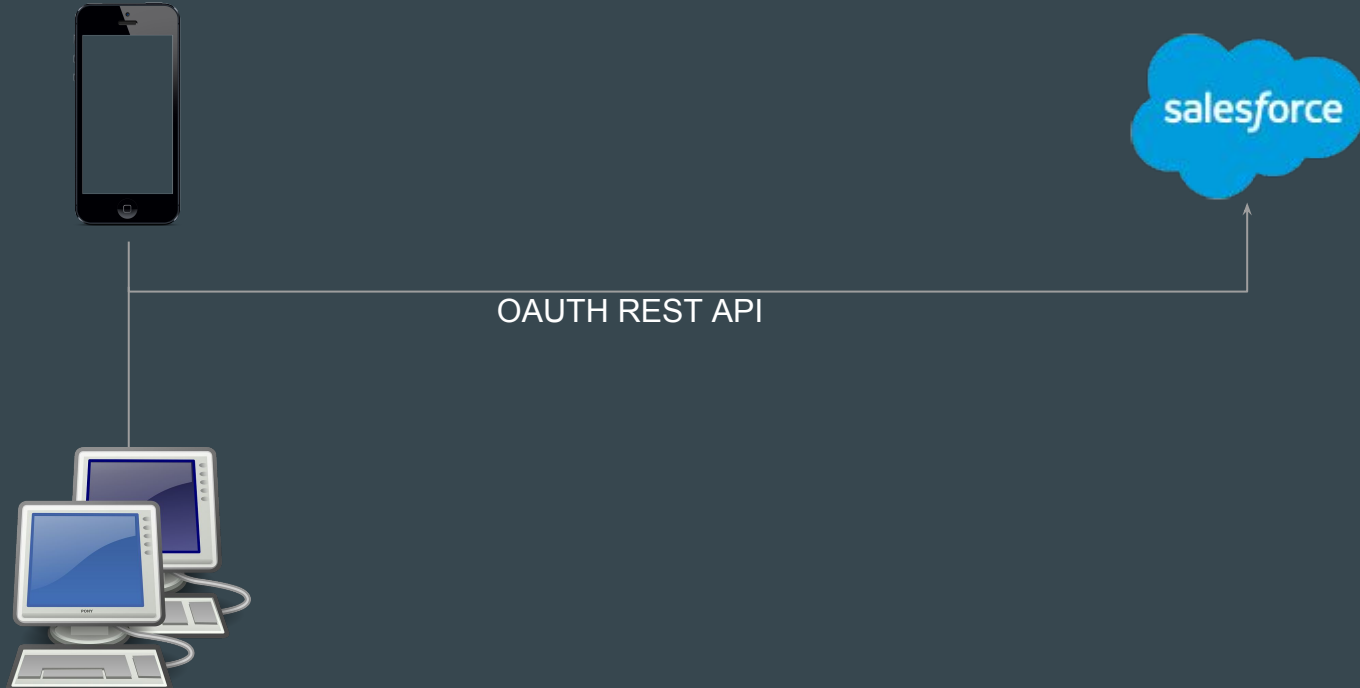
Swift

Architecture ...

Student Lifecycle Core Diagram (Technology View "As Is")
Nikki Rogers, Enterprise Architect, University of Bristol, April 2014




Mobile Architecture



Mobile App

Carrier 5:52 PM

Done login.salesforce.com



Username

Password

Log In

☐ Remember me

[Forgot Your Password?](#) [Use Custom Domain](#)

Carrier 5:49 PM

My Tasks Log Out

You have 3 tasks. Pull to refresh.

- Policy update (Status: Not Started) >
- Test task (Status: Not Started) >
- Call request (Status: Not Started)
United Oil & Gas Corp. >

Carrier 5:50 PM

< My Tasks Test task Save

Select task status

- Not Started ✓
- In Progress
- Completed
- Waiting on someone else
- Deferred

Intro REST API & HTTP

REST uses HTTP/S protocol to transfer data

HTTP Operations - GET POST PUT PATCH DELETE

Mostly data in text format represented by JSON

JSON is structured but free form - anything goes... (YAML, RAML, Apiary, Swagger, Mulesoft)

JSON File example

Standard & Custom API

Standard Salesforce API

SOAP API, **REST API**, Chatter REST API, Bulk API, Metadata API, Streaming API, Tooling

Mobile Apps mostly work with REST API (Representational State Transfer)

Standard REST API

<https://na35.salesforce.com/services/data/v36.0/subjects/contact/id>

Custom APEX REST API Setup

```
@RestResource(urlMapping='/FieldCase/*')  
global with sharing class RESTCaseController { ... }
```

URL Endpoint: <https://na35.salesforce.com/services/apexrest/FieldCase>

Authentication & Security

Salesforce using OAuth 2.0 protocol

All API clients must authenticate

Setup your app as Connected App

Get Security key to copy into mobile app source code

Ready to connect!

Tools & Mobile SDKs

Salesforce Mobile SDK 4.2 - open-source on github

<https://github.com/forcedotcom/SalesforceMobileSDK-iOS>

Swiftly Salesforce Framework <https://github.com/mike4aday/SwiftlySalesforce>

- PromiseKit (Version 3): "Not just a promises implementation, it is also a collection of helper functions that make the typical asynchronous patterns we use as iOS developers delightful too."
- Alamofire: "Elegant HTTP Networking in Swift"
- Locksmith: "A powerful, protocol-oriented library for working with the keychain in Swift."
- * [OMGHTTPURLQ](#) framework Vital extensions to NSURLRequest that Apple left out for some reason.

XCode 7.3.1 & Swift 2.2 (Apple Developer Program & [Swift.org](#))

Salesforce Developer ORG Signup: <https://developer.salesforce.com/signup>

Resources

github repo: <https://github.com/iandrosov/Swiftforce>

Force.com Platform API Documentation: https://developer.salesforce.com/page/Salesforce_APIs

Custom Apex REST API: https://developer.salesforce.com/page/Creating_REST_APIs_using_Apex_REST

Salesforce Mobile SDK: https://developer.salesforce.com/page/Mobile_SDK

RAML: <http://raml.org/> YAML <http://www.yaml.org/start.html>

Swagger <http://swagger.io/> Apiary <https://apiary.io/>

Blog Rest with APEX Service on iOS Native App - Objective-c

-<https://appirio.com/tech-blog/rest-with-apex-service-in-native-ios-apps>

CODE DEMO

&

Q&A